Delphi

บุญเลิศ เอี่ยมทัศนา

โปรแกรมด้วย Delphi : หน้าต่างกับการย่อไฟล์

นำหน้าต่าง SDI มาใช้ในการบีบย่อและขยายไฟล์ โดยให้รันโปรแกรม pkzip.exe และ pkunzip.exe อีกทอดหนึ่ง

มื่อกล่าวถึงเมนูแล้ว เรื่องที่ตามมาคงเป็นเรื่องของหน้าต่าง MDI เพื่อที่จะได้พูดถึงการสนธิเมนู แต่ก่อนถึงหน้าต่าง MDI ควรเป็นเรื่องของหน้าต่าง SDI ก่อน ทำให้ต้องเลยไปถึงไดอะล็อกบ็อกซ์ด้วย เพื่อการเปรียบเทียบ

หน้าต่าง SDI กับ MDI

ในเดลไฟเรียกหน้าต่างที่ไม่ใช่หน้าต่าง MDI ว่า SDI (เข้าใจว่าเพื่อให้เข้าคู่กัน) โดยที่

- หน้าต่าง MDI คือหน้าต่างในชุดของหน้าต่าง Multi Document Interface ประกอบด้วยหน้าต่างหลักและหน้าต่างรองจำนวนหนึ่ง เช่น หน้าต่าง File Manager มีหน้าต่าง ไฟล์ไดเรคทอรี หนึ่งหรือหลายบานเป็นหน้าต่างรองอยู่ภายใน

- หน้าต่าง SDI คือหน้าต่างในชุดของหน้าต่าง Single Document Interface ประกอบด้วยหน้าต่างหลัก 1 บาน และอาจมีหน้าต่างรองบานหนึ่งหรือหลายบานหรือไม่มี เช่น หน้าต่างของโปรแกรมโดยทั่วไป ซึ่งก็คือหน้าต่างตามที่ผมเคยกล่าวมาทั้งหมด

ในสองฉบับนี้

ดังที่ในสองฉบับนี้จะนำหน้าต่าง SDI มาใช้ในการบีบย่อและขยายไฟล์ โดยให้รันโปรแกรม pkzip.exe และ pkunzip.exe อีกทอดหนึ่ง

ทั้งนี้เพื่อการศึกษาเปรียบเทียบด้วยว่า เมื่อเขียนโปรแกรมด้วยวิชวลเบสิก (บทความเรื่อง Visual Basic : เขียนฟรอนเอ็นสำหรับ PKZIP บนวินโดวส์ ของคุณ ขจร โรจนเมธินทร์ ในไมโครฯ 123) เขียนอย่างไร และเมื่อเขียนด้วยเดลไฟเขียนอย่างไร

การใช้งานหน้าต่าง SDI

ผมจำแนกการใช้งานหน้าต่าง SDI ที่เป็นหน้าต่างรองออกได้เป็น 3 รูปแบบ และขอนำชื่อหน้าต่างของโปรแกรมนี้มาอ้างถึงด้วย กล่าวคือ

1. เป็นหน้าต่างเอกเทศ ที่ทำงานของตนโดยไม่เกี่ยวข้องกับหน้าต่างอื่น แต่อาจรับการสนับสนุนจากหน้าต่างอื่น (หน้าต่าง UnZip Form, Bitmap Window และ Text Window)

2. เป็นหน้าต่างสนับสนุน เพื่อสนับสนุนการทำงานของหน้าต่างอื่น (หน้าต่าง File Directory Box)

3. เป็นหน้าต่างอย่างไดอะล็อกบ็อกซ์ คือ มีลักษณะการทำงานอย่างไดอะล็อกบ็อกซ์ แต่ใช้งานในฐานะหน้าต่าง (หน้าต่าง Zip Form)

การโปรแกรม

เมื่อต้องการใช้หน้าต่างหลายบาน วิธีการตรงตัวครับคือ ด้วยการเปิดฟอร์มใหม่ (โดยเลือกรายการ File/New Form) หลายฟอร์ม เพราะฟอร์มคือหน้าต่าง ซึ่งเมื่อเปิดฟอร์มใหม่จะได้ทั้งฟอร์มและยูนิต เช่น ได้เป็น Form2 กับ Unit2 (ของหน้าต่างหลักคือ Form1 กับ Unit1) และจะขออ้างชื่อตามนี้ด้วย เพื่อดำเนินการต่อไป คือ

1. เก็บลงไฟล์ เมื่อเปิดฟอร์มใหม่แล้วควรเก็บลงไฟล์ โดยเลือกรายการรองใน File แยกเป็น

- เก็บพร้อมกับโปรเจ็กต์ เลือกรายการ Save Project As หรือต่อไปคือ Save Project
- เก็บโดยตรง สามารถเก็บไฟล์ .pas ได้โดยตรง โดยเลือกที่ฟอร์มนั้นให้แอคตีฟแล้วเลือกรายการ Save File As และต่อไปคือ Save File

ถ้าเป็นไฟล์ใหม่จะถูกถามชื่อไฟล์ .pas (เพื่อใช้แทน unit2.pas) ซึ่งควรอย่างยิ่งที่จะให้ชื่อใหม่ เช่น ให้เป็น UFIBOX2.PAS จะทำให้ได้ชื่อยูนิตเป็น Ufibox2

2. เตรียมโปรแกรม เช่นเดียวกับของหน้าต่างหลักคือ กำหนดคอมโปเนนต์ในฟอร์ม และป้อนโปรแกรมในยูนิต

3. นำไปใช้ การนำไปใช้อาจหมายถึงเพียงแค่ให้ฟอร์มใหม่แสดงตัว หรืออาจประกอบด้วยการใช้คำในยูนิตของฟอร์มใหม่ด้วย ซึ่งอาจใช้ฟอร์มหนึ่งในฟอร์มใดก็ได้และใช้ได้ในหลายฟอร์ม ขอสมมติว่าจะใช้ Form3 (Unit3) ใน Form4 (Unit4) จะใช้ได้ดังนี้

- บ่งใช้ Unit3 ใน uses ของ Unit4 (ที่อยู่ใต้คำว่า Interface)
- ให้แสดงตัวด้วย Show
- ใช้คำใน Unit4 ตามต้องการ (คำคือ แบบของข้อมูลรวมสกุลโปรแกรมย่อย ค่าคงตัว ตัวแปรรวมออปเจ็กต์)

แต่ถ้าใน Unit3 ต้องการใช้คำใน Unit4 จะต้องบ่งใช้ Unit4 ใน Unit3 เป็นการบ่งใช้ยูนิตไขว้กัน ให้เพิ่มประโยค uses ใต้คำว่า Implementation ใน Unit3 แล้วบ่งใช้ Unit4 ใน uses นี้

4. การถอนและผนวกฟอร์ม การเปิดฟอร์มตามที่กล่าวมานี้ จะเป็นการผนวกฟอร์มเข้ากับโปรเจ็กต์ด้วย ต่อไปหากต้องการ ให้เลือกรายการ View/Project Manager จะแสดงหน้าต่างโปรเจ็กต์ เพื่อ

- เลือกชื่อฟอร์ม แล้วเลือกปุ่ม Remove เพื่อถอนออกจากโปรเจ็กต์
- เลือกปุ่ม Add เพื่อผนวกฟอร์มที่สร้างไว้แล้ว เช่นของโปรเจ็กต์อื่น
- เลือกชื่อฟอร์ม แล้วเลือกปุ่ม View unit หรือ View form เพื่อให้แสดงตัวตามที่จะกล่าวต่อไป

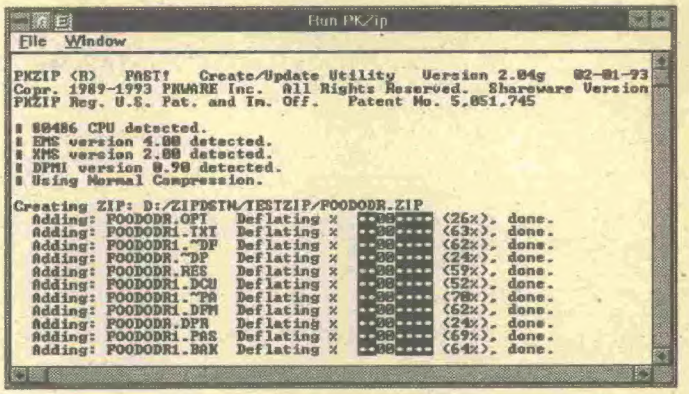
5. ให้แสดงตัว เมื่อเตรียมโปรแกรมแล้วอาจปิดหน้าต่างฟอร์ม หรือเมื่อเปิดโปรเจ็กต์นี้ใหม่ในครั้งต่อไป ซึ่งจะแสดงฟอร์มและยูนิตของหน้าต่างหลักเท่านั้น หรือเมื่อหน้าต่างบังกัน จะให้แสดงหน้าต่างฟอร์มและยูนิตที่ต้องการได้โดย

- ตามข้อ 4 หรือ
- เลือกรายการ View/Units หรือ Forms แล้วเลือกชื่อยูนิตหรือฟอร์มที่ต้องการ

6. หน้าต่างเปิดหรือหน้าต่างปิด การเปิดฟอร์มตามที่กล่าวมานี้ จะได้เป็นหน้าต่างเปิด (Auto-create forms) อาจทำเป็นหน้าต่างปิด (Available forms) ได้ ซึ่งจะต้องให้กำเนิดก่อนนำมาใช้ จึงจะขอนำไปกล่าวภายหลัง

โปรแกรม RunPKZip

โปรแกรม RunPKZip เป็นโปรแกรมสำหรับรันโปรแกรม PKZIP และ PKUNZIP เพื่อการย่อและขยายไฟล์ จึงต้องมีไฟล์ PKZIP.EXE และ PKUNZIP.EXE ในไดเรคทอรีที่บ่งชื่อไว้ใน PATH= ใน AUTOEXEC.BAT



รูปที่ 1 หน้าต่างหลักของโปรแกรม RunPKZip แสดงข้อความที่เป็นรายงานการย่อไฟล์

เมื่อรันโปรแกรมนี้จะแสดงหน้าต่างหลักดังในรูปที่ 1 ที่รายการ Window มีรายการรองให้เลือกดังนี้

1. Zip จะเปิดหน้าต่าง Zip Form เพื่อการย่อไฟล์ไว้ในไฟล์ .ZIP
2. Un Zip จะเปิดหน้าต่าง UnZip Form เพื่อการดู (view) ข้อมูลอันเกี่ยวกับการย่อไฟล์ในไฟล์ .ZIP หรือขยายไฟล์ในไฟล์ .ZIP
3. File Box จะเปิดหน้าต่าง File Box คือ File Directory Box เพื่อการเปิดไฟล์ ซึ่งความจริงแล้วเป็นงานรอง งานหลักคือสนับสนุนหน้าต่าง Zip Form และ UnZip Form ในเรื่องของชื่อไฟล์ และไดเรคทอรี

ดังจะกล่าวในรายละเอียดในแต่ละหน้าต่างต่อไป

ลิสติงที่ 1

```
PROGRAM  RunPKZip;  { runpkzip.dpr }

Form == Form1 ( Run PKZip )
Unit == pkmain1  { pkmain1.pas }
        - uses = + UFiBox2,UZipWnd3,UUnZWnd4

Components :-
1 Standard :: MainMenu == MainMenu1
  3.0 Main = ( &File, &Window )
  3.1 File = ( &New, &Open..., Save &As..., -, E&xit )
  3.1 Window = ( &Zip, &Un Zip, &File Box )
      - GroupIndex = 9
2 Standard :: Memo == Memo1
  - Align = alClient
  - Font.Name = Terminal
  - Font.Size = 9
  - ScrollBars = ssBoth
3 Dialogs :: OpenDialog == OpenDialog1
  - DefaultExt = txt
  - Filename = *.txt
  - Filter = [ Text files (*.txt), All files (*.*) ]
  - Options =  [ ofPathMustExist(T), ofFileMustExist(T) ]
4 Dialogs :: SaveDialog == SaveDialog1
  - DefaultExt = txt
  - Filename = *.txt
  - Filter = [ Text files (*.txt), All files (*.*) ]
  - Options =  [ ofOverwritePrompt(T), ofPathMustExist(T) ]

(* -------- Implementation part -------- *)

procedure TForm1.Exit1Click(Sender: TObject);
begin
  Close;
end;

(* --------- *)

procedure TForm1.New1Click(Sender: TObject);
begin
  Memo1.Clear;
end;

procedure TForm1.Open1Click(Sender: TObject);
begin
  if  OpenDialog1.Execute  then
    Memo1.LineS.LoadFromFile(OpenDialog1.FileName);
end;

procedure TForm1.SaveAs1Click(Sender: TObject);
begin
  if  SaveDialog1.Execute  then
    Memo1.LineS.SaveToFile(SaveDialog1.FileName);
end;

(* ------ *)


procedure TForm1.Zip1Click(Sender: TObject);
begin
  ZipWnd3.Show;  FiBox2.Show;
end;

procedure TForm1.UnZip1Click(Sender: TObject);
begin
  UnZWnd4.Show;  FiBox2.Show;
end;

procedure TForm1.FileBox1Click(Sender: TObject);
begin
  FiBox2.Show;
end;

end.
```

หน้าต่างหลัก

ให้เตรียมโปรแกรมของหน้าต่างหลักตามรูปที่ 1 และตามลิสติงที่ 1 โดยที่

1. ได้ให้บ่งใช้ยูนิต 3 ยูนิต แต่ขอให้บ่งใช้เมื่อได้เตรียมฟอร์มของยูนิตนั้นแล้ว

2. เช่นกันที่ท่อนท้ายมีกิจกรรมอยู่ 3 กิจกรรม เพื่อการเปิดใช้หน้าต่างต่าง ๆ ด้วยการให้แสดงด้วย Show ซึ่งจะป้อนได้ หมายถึงเพื่อการคอมพายล์และรันทดสอบ เมื่อได้เตรียมโปรแกรมของหน้าต่างนั้น ๆ แล้ว

3. กิจกรรมที่เหลือ เป็นกิจกรรมสำหรับลบ อ่าน และเขียนกับไฟล์ กับข้อมูลในกรอบ Memo1 และยุติการทำงานด้วย Close

4. เก็บลงไฟล์โดยให้ชื่อไฟล์ .pas เป็น ufibox2.pas จะได้ชื่อยูนิตเป็น Ufibox2 แต่จะอ้างถึงเป็น UFiBox 2

ทั้งนี้หมายถึงได้เก็บหน้าต่างหลักคือโปรเจ็กต์ลงไฟล์ก่อนแล้วหรือพร้อมกัน

การเตรียมฟอร์มใหม่

โดยที่ในโปรแกรมตามที่จะกล่าวต่อไปจะต้องเปิดฟอร์มใหม่หลายฟอร์ม จึงขอนำกรรมวิธีที่เหมือนกันมากล่าวก่อนคือ เมื่อจะให้เปิดฟอร์มใหม่ ดังเช่นเมื่อต้องการให้เปิดฟอร์มใหม่สำหรับหน้าต่าง File Box จะให้คำอธิบายว่า

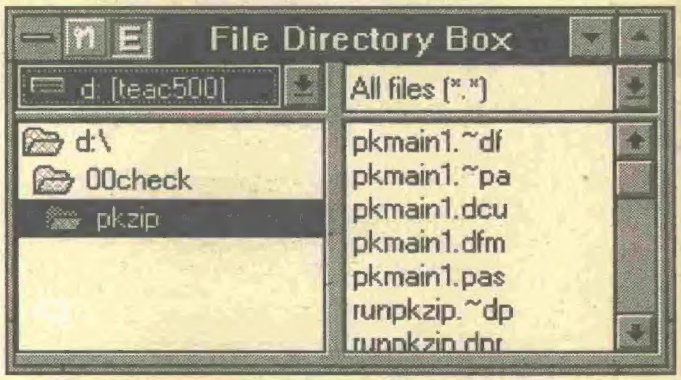
"ให้เตรียมฟอร์มใหม่ตามลิสติงที่ 2 ได้เป็นฟอร์ม FiBox2 และยูนิต UFiBox2" (ชื่อฟอร์มและชื่อยูนิต มีอยู่ในลิสติง นำมาอ้างถึงเพื่อเน้น)

จากคำอธิบายนี้หมายถึงให้ดำเนินการอย่างน้อยที่สุด (คือยังไม่ต้องเตรียมโปรแกรมทั้งหมด) ดังนี้

1. เปิดฟอร์มใหม่

2. ให้ค่า Caption ตามลิสติง กรณีนี้คือ File Directory Box

3. ให้ค่า Name เป็น FiBox 2 ซึ่งจะทำให้เปลี่ยนชื่อฟอร์มจาก Form2 เป็น FiBox2 และเปลี่ยนชื่อสกุลจาก TForm2 เป็น TFiBox2



รูปที่ 2 หน้าต่าง File Box

หน้าต่าง File Box

เมื่อเลือกรายการ Window/File Box ในหน้าต่างหลัก จะแสดงหน้าต่าง File Box ดังรูปที่ 2 เพื่อให้

- แดร็ก ชื่อไดเรคทอรี หรือชื่อไฟล์ ไปดร็อปในหน้าต่างอื่น (ตามที่จะกล่าวต่อไป) ซึ่งจะขอเรียกเป็นการดร็อปชื่อไดเรคทอรีหรือชื่อไฟล์

(การแดร็กชื่อไดเรคทอรีจะต้องเปิดไดเรคทอรีนั้นก่อน โดยดับเบิ้ลคลิ๊ก)

ลิสติงที่ 2

```
{ program RunPKZip }

Form == FiBox2 { File Directory Box }
        - Font.Name = MS Sans Serif
        - Font.Size = 8
        - FormStyle = fsStayOnTop
        - PopupMenu = PopupMenu1
Unit == UFiBox2  { ufibox2.pas }
        - uses = + UBmpWnd5,UTxtWnd6

Components :-
1 System :: DriveComboBox == DriveComboBox1
  - DirList = DirectoryListBox1
2 System :: DirectoryListBox == DirectoryListBox1
  - DragMode = dmAutomatic
  - FileList =  FileListBox1
```

(ต่อ)

ลิสติงที่ 2 (ต่อ)

```
3 System :: FilterComboBox == FilterComboBox1
  - FileList =  FileListBox1
  - Filter = [ All files(*.*), Bitmap files(*.bmp) ]
4 System :: FileListBox == FileListBox1
  - DragMode = dmAutomatic
  - PopupMenu = PopupMenu1
5 Standard ::  PopupMenu ==  PopupMenu1
  5.0 Popup = ( &Run, Open &Bitmap, Open &Text )

(* -------- Declaration part -------- *)

type
  TFiBox2 = class(TForm)
    ..................
  private
    { Private declarations }
  public
    { Public declarations }
    function  DragFromFiBox(oSrce : TObject) : Boolean;
    function  GetDirItem(var txtS : String;
                         oSrce : TObject) : Boolean;
  end;

var
  FiBox2: TFiBox2;

function  RunBatFi(const cmmLnS,logFiNmeS : OpenString;
                         oStrS : TStringS) : Boolean;

(* -------- Implementation part -------- *)

function  RunBatFi(const cmmLnS,logFiNmeS : OpenString;
                         oStrS : TStringS) : Boolean;
var  winMd : Word;  cmLneB : array[0..255] of Char;
begin
  Application.ProcessMessages ;
  winMd := WinExec(StrPCopy(cmLneB, cmmLnS), sw_Show);
  if  winMd < 32  then
    begin  Result := False;  MessageBeep(0);  Exit  end;
  Result := True;
  while  GetModuleUsage(winMd) > 0  do
    Application.ProcessMessages ;
  if  oStrS <> nil  then
   oStrS.LoadFromFile(logFiNmeS);
end;

(* -------- *)

function  TFiBox2.DragFromFiBox(oSrce : TObject) : Boolean;
begin
  Result := (oSrce is TDirectoryListBox) or
            (oSrce is TFileListBox)
end;


function  TFiBox2.GetDirItem(var txtS : String;
                             oSrce : TObject) : Boolean;
begin
  if  (oSrce is TDirectoryListBox)  then
    begin  txtS := DirectoryListBox1.Directory;
           Result := True  end
  else
    begin  txtS := FileListBox1.FileName;
           Result := False  end
end;


(* ------------ *)
```

(ต่อ)

ลิสติงที่ 2 (ต่อ)

```
procedure TFiBox2.Run1Click(Sender: TObject);
var  txtS : String;
begin
  txtS := FileListBox1.FileName;
  if  ExtractFileExt(txtS) <> '.exe'  then
    begin  ShowMessage('Can''t Open this file'); Exit  end;
  TxtS[Length(txtS)+1] := #0;
  WinExec(@txtS[1], sw_Show)
end;


procedure TFiBox2.OpenBitmap1Click(Sender: TObject);
var  txtS : String;
begin
  txtS := FileListBox1.FileName;
  BmpWnd5.Image1.Picture.LoadFromFile(txtS);
  BmpWnd5.Show
end;


procedure TFiBox2.OpenText1Click(Sender: TObject);
var  txtS : String;
begin
  txtS := FileListBox1.FileName;
  TxtWnd6.Memo1.LineS.LoadFromFile(txtS);
  TxtWnd6.Show;
end;


end.
```

- เลือกชื่อไฟล์ .exe หรือไฟล์รูปภาพ หรือเท็กซ์ไฟล์ แล้วคลิ๊กเม้าส์ขวาเพื่อเปิดเมนูลอยเพื่อเลือกรายการ Run หรือ Open Bitmap หรือ Open Test เพื่อการรันหรือให้แสดงรูปภาพ หรือข้อความ ตามลำดับ

เตรียมฟอร์ม

โดยที่หน้าต่าง File Box จะมีหน้าต่างบริวารอีก 2 หน้าต่าง จึงขอให้เตรียมฟอร์มใหม่ ดังนี้

1. ตามลิสติงที่ 2 ได้เป็นฟอร์ม FiBox2 และยูนิต UFiBox2 เพื่อใช้เป็นหน้าต่าง File Box

ลิสติงที่ 3

```
{ program RunPKZip }

Form == BmpWnd5 ( Bitmap Window )
Unit == UBmpWnd5  ( ubmpwnd5.pas )

Components :-
1 Additional :: Image == Image1
  - AutoSize := True

end.
```

2. ตามลิสติงที่ 3 และ 4 ได้เป็นฟอร์ม BmpWnd5 และ TxtWnd6 กับยูนิต UBmpWnd5 และ UTxtWnd6 เพื่อใช้เป็นหน้าต่าง Bitmap Window และ Text Window ซึ่งทั้งคู่นี้มีคอมโปเนนต์เพียงหนึ่งเดียว และไม่ต้องป้อนโปรแกรม จึงให้ กำหนดคอมโปเนนต์ เลื่อนไปไว้ตรงมุมบนซ้าย เก็บลงไฟล์ แล้วปิดหน้าต่าง (เพื่อไม่ให้เกะกะ)

เตรียมโปรแกรม

เตรียมโปรแกรมตามรูปที่ 2 และตามลิสติงที่ 2 โดยกำหนดคอมโปเนนต์เป็น

ลิสติงที่ 4

```
{ program RunPKZip }

Form == TxtWnd6 ( Text Window )
Unit == UTxtWnd6  ( utxtwnd6.pas )

Components :-
1 Standard :: Memo == Memo1
  - Align = alClient
  - Font.Name = Treminal
  - Font.Size = 9
  - ScrollBars = ssBoth
end.
```

- DriveComboBox1 และ DirectoryListBox1 ทางซีกซ้าย
- FilterComboBox1 และ FileListBox1 ทางซีกขวา
- PopupMenu1

ให้ค่าพรอพเพอร์ตี้ส์ และป้อนโปรแกรมทั้งหมด

คำอธิบาย

ในโปรแกรมของหน้าต่างนี้มีโปรแกรมอยู่ 3 เรื่องคือ การรันไฟล์ .bat การแดร็กแอนด์ดรอป และโปรแกรมของเมนูลอย ตามที่จะกล่าวในแต่ละหัวข้อรองต่อไปนี้ คือ

การรันไฟล์ .bat

โดยที่โปรแกรม PKZIP และ PKUNZIP จะรายงานผลของการย่อและขยายไฟล์ออกทางจอภาพตามวิธีของโปรแกรมที่รันบน DOS ซึ่งผู้ใช้อาจต้องการผลนี้ จะทำได้โดย redirection เอาต์พุตนี้ไปเก็บไว้ในไฟล์และนำออกแสดงในภายหลัง ด้วยคำสั่งของ DOS ในไฟล์ . bat ดังที่ได้ทำเป็นฟังก์ชัน RunBatFi (ไว้เป็นโกลบอล เพราะจะมีการรันไฟล์ .bat ในที่หลายแห่ง) กล่าวคือ

1. การผ่านค่า ได้กำหนดให้ผ่านค่ามาให้ตามลำดับดังนี้

- cmmLnS เป็นค่าคอมมานด์ไลน์อันประกอบด้วยชื่อไฟล์ .bat และค่าพารามิเตอร์ที่จะให้แก่ไฟล์ .bat เพื่อให้แก่ PKZIP หรือ PKUNZIP อีกทอดหนึ่ง
- LogFiNmes ชื่อไฟล์ (เป็นเท็กซ์ไฟล์) ที่จะให้เก็บรายงานการย่อหรือขยายไฟล์
- oStrS เป็นออปเจ็กต์ของสกุล TStringS เช่น LineS ในกรอบ Memo หรือ Items ในกรอบ ListBox

2. โปรแกรม ได้ให้ดำเนินการดังนี้

- ให้ ProcessMessage ถอนเมสเสจจากคิว เพราะอาจมีเมสเสจตกค้าง
- ให้ฟังก์ชัน WinExec รันโปรแกรมในคอมมานด์ไลน์ กรณีนี้คือไฟล์ .bat ซึ่งจะให้ค่ากลับมาเป็นหมายเลขโมดูล (instance of loaded module) แต่ถ้ามีปัญหาจะให้ค่าน้อยกว่า 32
- ฟังก์ชัน GetModuleUsage จะให้ค่าหมายเลขก๊อบปี้ของโมดูล เช่น ถ้าเปิดใช้ Notepad สามหน้าต่าง ก็จะได้หมายเลขก๊อบปี้เป็น 1, 2 และ 3 ตามลำดับ ซึ่งเมื่อปิดหน้าต่างบานใด และสำหรับโปรแกรมบน DOS คือเมื่อยุติการรัน จะได้หมายเลขก๊อบปี้เป็น 0
- ในโปรแกรมจึงใช้วงรอบ while ตรวจหมายเลขก๊อบปี้ ซึ่งต้องใช้คู่กับ ProcessMessage โปรแกรมจึงติดอยู่ตรงนี้จนกว่าโปรแกรมใน DOS จะยุติการรัน
- จึงจะทำคำสั่งต่อไป ซึ่งให้ oStrS โหลดข้อความในไฟล์ logFiNmeS ซึ่งถ้า oStrS เป็นของกรอบ Memo หรือกรอบ ListBox จะนำออกแสดง

เรื่องนี้ขอขอบคุณ คุณชจร โรจนเมธินทร์ ครับ ที่นำการใช้ฟังก์ชัน GetModuleUsage ของวินโดวส์มาแสดง เพราะผมเองไม่เคยคิดที่จะใช้ ของเดิมผมให้แสดงเมสเสจบ็อกซ์ตรง While ซึ่งจะเห็นต่อเมื่อโปรแกรมใน DOS ยุติการทำงานแล้ว ได้ผลเช่นเดียวกันแต่ใช้ฟังก์ชัน GetModuleUsage ดีกว่า

การแดร็กแอนด์ดร็อป

จากการเตรียมโปรแกรมคือที่ให้ค่า DragMode เป็น dmAutomatic เท่ากับกำหนดให้ผู้ใช้สามารถแดร็กจากกรอบรายการไดเรคเทอรีคือ DirectoryListBox1 หรือจากกรอบรายการไฟล์คือ FileListBox1 ได้โดยอัตโนมัติ (เราไม่ต้องเขียนโปรแกรมเสริม) จึงได้เตรียมโปรแกรมสำหรับสนับสนุนการทำงานในเรื่องนี้ดังนี้

1. กิจกรรม DragFormFiBox เป็นฟังก์ชันเพื่อให้ตรวจว่า ได้แดร็กไปจากกรอบรายการไดเรคทอรี หรือกรอบรายการไฟล์ หรือไม่
2. กิจกรรม GetDirItem จะให้ชื่อไดเรคทอรี หรือชื่อไฟล์ ที่ผู้ใช้เลือกไว้ก่อนแดร็กแก่ txtS ซึ่งถ้าเป็นชื่อไดเรคเทอรีจะให้ตนเองเป็นจริง

โปรแกรมของเมนูลอย

ได้จัดกิจกรรมไว้สำหรับรายการในเมนูลอยดังนี้

1. Run1Click (รายการ Run) เพื่อการรันโปรแกรมในไฟล์ .exe
2. OpenBitmap1Click (รายการ Open Bitmap) เพื่อแสดงภาพในไฟล์ .bmp (รวม ico และ wmf) ในหน้าต่าง Bitmap Window
3. OpenText1Click (รายการ Open Text) เพื่อแสดงข้อความในเท็กซ์ไฟล์ ในหน้าต่าง Text Window

จุดประสงค์นี้เพื่อใช้ตรวจดูผลของการขยายไฟล์

สำหรับการสร้างเมนูลอย เช่นเดียวกับการสร้างเมนูหลักตามที่กล่าวในไมโครฯ 126 โดยสังเขปซึ่งจะต้องดำเนินการต่อเนื่องดังนี้

- กำหนดคอมโปเนนต์เป็น PopupMenu1 แล้วดับเบิ้ลคลิ๊กที่คอมโปเนนต์นี้
- ป้อนข้อความที่เป็นรายการแล้วกดคีย์ Enter ทั้งสามรายการ
- ดับเบิ้ลคลิ๊กที่รายการเพื่อป้อนโปรแกรม

หากต้องการทดลองรันโปรแกรมให้ บ่งใช้ยูนิต UFiBox2 ในยูนิต PKMain1 และป้อนโปรแกรมของกิจกรรม FileBox1Click ตามลิสติงที่ 1

หน้าต่าง Zip Form

เมื่อเลือกรายการ Window/Zip ในหน้าต่างหลัก จะแสดงหน้าต่าง Zip Form และหน้าต่าง File Box ดังในรูปที่ 2 เพื่อการบีบย่อไฟล์ กล่าวคือ

วิธีใช้

การบีบย่อไฟล์ด้วยโปรแกรม PKZIP และตามที่จัดไว้ในโปรแกรมนี้ จะเลือกดำเนินการได้ดังนี้

1. ไฟล์ต้นทางที่จะนำมาย่อ เลือกได้เป็น

- ไฟล์เดี่ยว โดยดร็อปชื่อไฟล์
- ไดเรคทอรี โดยดร็อปชื่อไดเรคทอรี

ซึ่งทั้งคู่นี้ให้ดร็อปในกรอบ Edit1 คือกรอบ Zip form dir or file (s)

2. ถ้าเป็นไดเรคทอรี ให้เลือกไวล์การ์ดในหน้า Include หรือ Exclude (หน้าใดหน้าหนึ่ง) ไว้ก่อนดร็อปชื่อไดเรคทอรี กล่าวคือ

- หน้า Include เพื่อให้ย่อไฟล์เฉพาะนามสกุลที่เลือก ส่วน Exclude เว้นนามสกุลที่เลือก
- สามารถเลือกนามสกุลในหน้าเดิมหรือป้อนแก้โดยตรง
- หากต้องการให้ย่อไฟล์ในไดเรคทอรีรองด้วยให้เซตปุ่ม Sub-Directory และถ้าต้องการให้บันทึกไดเรคทอรีให้เซตปุ่ม Include path

3. ไฟล์ปลายทาง ที่จะเก็บผลการย่อ ซึ่งไม่ว่าจะย่อไฟล์เดียวหรือทั้งไดเรคทอรี จะเก็บผลไว้ไฟล์ .ZIP เพียงไฟล์เดียว ให้ ดร็อปชื่อไดเรคทอรี หรือชื่อไฟล์ .ZIP (เดิม) ในกรอบ Edit2 คือกรอบ Zip to file ซึ่งถ้าเป็นชื่อไดเรคทอรีเพื่อป้อนชื่อไฟล์ .ZIP ต่อท้าย (ไม่จำเป็นต้องป้อนนามสกุล)

4. เลือกการบีบย่อในรายละเอียดในกรอบต่าง ๆ ได้อีกดังนี้

4.1 Add เลือกได้เป็น

- Add ผนวกต่อท้าย
- Freshen ทับของเดิม (หากของใหม่ใหม่กว่า)
- UpDate ทับของเดิมและผนวกต่อท้ายหากมีไฟล์ใหม่

4.2 Method วิธีที่ใช้ในการย่อ ไล่จาก Normal คือหลายครั้งถึง Direct Copy ไม่มีการย่อ

4.3 Misc ที่เหลือคือ

- Move ลบไฟล์ต้นทาง
- Span disk เมื่อต้องการย่อไฟล์ลงแผ่นดิสค์หลายแผ่น

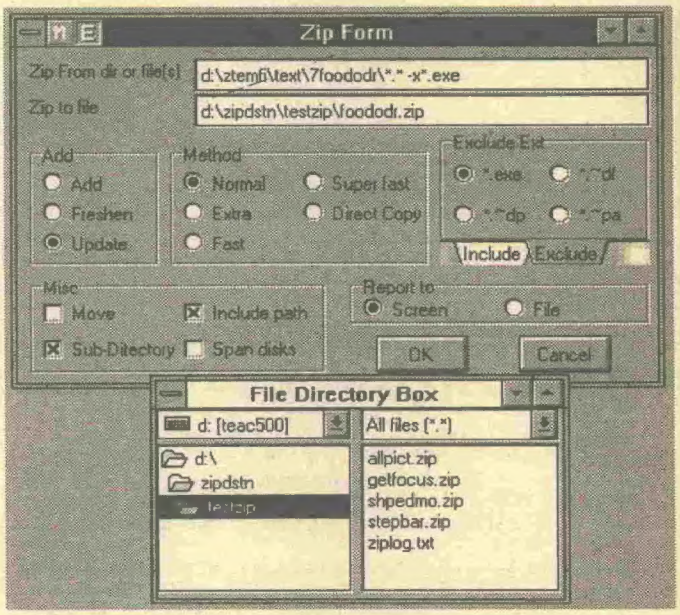
4.4 Report to เพื่อให้รายงานการย่อไฟล์ ทาง

- Screen จอภาพ (ขอแนะนำให้ใช้วิธีนี้เป็นหลัก)
- File ทางไฟล์ ซึ่งจะอ่านมาแสดงในหน้าต่างหลัก

เมื่อได้เซตค่าต่าง ๆ แล้วให้เลือกปุ่ม OK ซึ่งถ้าให้ออกทางจอภาพอาจดูผลไม่ทัน จะดูได้ด้วยหน้าต่าง View Form ตามที่กล่าวต่อไป

เตรียมโปรแกรม

เตรียมฟอร์มใหม่ตามลิสติงที่ 5 ได้เป็นฟอร์ม ZipWnd3 และยูนิต UZipWnd3 เพื่อเตรียมโปรแกรมตามรูปที่ 3 และตามลิสติงที่ 5 กล่าวคือ



รูปที่ 3 หน้าต่าง Zip Form และ File Box ในหน้าต่าง Zip Form แสดงว่าได้เลือกไวล์การ์ดไว้เป็น *exe ในหน้า Exclude แล้วแดร็กชื่อไดเรคทอรี (7 foododr) มาดร็อปในกรอบ Edit1และได้แดร็กชื่อไดเรคทอรี (Testzip) มาดร็อปในกรอบ Edit2 เพื่อป้อนชื่อไฟล์เป็น foododr.zip

ลิสติงที่ 5

```
{ program RunPKZip }

Form == ZipWnd3 ( Zip Form )
        - Font.Name = MS Sans Serif
        - Font.Size = 8
Unit == UZipWnd3  ( uzipwnd3.pas )
        - uses = + UFiBox2

Components :-
1 Standard :: Label == Label1 ( Zip from dir or file(s) )
```

(ต่อ)

ลิสติงที่ 5 (ต่อ)

```
  2 Standard :: Edit == Edit1
  3 Standard :: Label == Label2 ( Zip to file )
  4 Standard :; Edit == Edit2
    - Text = c:\untitle.zip
  5 Standard :: RadioGroup == rgAdd ( Add )
    - ItemIndex = 2
    - Items = ( Add, Freshen, Update )
  6 Standard :: RadioGroup == rgMethod ( Method )
    - Columns = 2
    - ItemIndex = 0
    - Items = ( Normal, Extra, Fast, Super fast, Direct Copy )
  7 Additional :: NoteBook == nbSelExt
    - PageIndex = 0
    - Pages = ( Include, Exclude )
    7.1 Include :-
        7.1.1 Standard :: RadioGroup == rgInclExt ( Include Ext )
              - Columns = 2
              - ItemIndex = 2
              - Items = ( *.*, *.txt, Delphi, *.bmp, *.pas )
    7.2 Exclude :-
        7.2.1 Standard :: RadioGroup == rgExclExt ( Exclude Ext )
              - Columns = 2
              - Items = ( *.exe, *.~dp, *.~df, *.~pa )
  8 Additional :: TabSet == tsSelExt
  9 Standard :: GroupBox == gbMisc ( Misc )
    Components :-
    9.1 Standard :: CheckBox == CheckBox1 ( Move )
        - Tag = 109
    9.2 Standard :: CheckBox == CheckBox2 ( Include path )
        - Checked = True
        - Tag = 112
    9.3 Standard :: CheckBox == CheckBox3 ( Sub-Directory )
        - Checked = True
        - Tag = 114
    9.4 Standard :: CheckBox == CheckBox4 ( Span disks )
        - Tag = 38
 10 Standard :: RadioGroup == rgReport ( Report to )
    - ItemIndex = 0
    - Items = ( Screen, File )
 11 Standard :: Button == bttOK ( OK )
 12 Standard :: Button == bttCancel ( Cancel )

(* -------- Declaration part -------- *)

type
  TZipWnd3 = class(TForm)
    ................
  private
    { Private declarations }
    oRprtStrS : TStringS;
    function  GetOptStr : String;
  public
    { Public declarations }
  end;

(* ------- Implementation part ------- *)
implementation
uses PkMain1;

const  ZipBatFiNmeS = 'c:\dozip.bat';
       ZipLogFiNmeK = 'c:\ziplog:txt';

procedure TZipWnd3.FormCreate(Sender: TObject);
begin
  rgReportClick(Sender);
  tsSelExt.Tabs := nbSelExt.Pages;
end;

procedure TZipWnd3.FormDestroy(Sender: TObject);
```

(ต่อ)

ลิสติงที่ 5 (ต่อ)

```
begin
  { DeleteFile(zipBatFiNmeS);
    DeleteFile(ZipLogFiNmeK);  }
end;

procedure TZipWnd3.bttCancelClick(Sender: TObject);
begin
  Close
end;

procedure TZipWnd3.tsSelExtClick(Sender: TObject);
begin
  nbSelExt.PageIndex := tsSelExt.TabIndex;
end;

(* ------- *)
const
  RprtToScrnP : PChar =
   '@echo off' + #13#10 +
   'pkzip %1 %2 %3 %4 %5 %6 %7 %8 %9' +
   #13#10 + 'exit';

  RprtToFiP : PChar =
   '@echo off' + #13#10 +
   'pkzip %1 %2 %3 %4 %5 %6 %7 %8 %9 > ' + ZipLogFiNmeK +
   #13#10 + 'exit';

procedure TZipWnd3.rgReportClick(Sender: TObject);
var  oBatFi : TStringList;  P : PChar;
begin
  bttOK.Enabled := False;
  case  rgReport.ItemIndex  of
    0 : begin  P := RprtToScrnP;
               oRprtStrS := nil  end;
    1 : begin  P := RprtToFiP;
               oRprtStrS := Form1.Memo1.LineS  end;
  end;  { case }
  oBatFi := TStringList.Create;
  oBatFi.SetText(P);  oBatFi.SaveToFile(zipBatFiNmeS);
  bttOK.Enabled := True;  oBatFi.Free;
end;

(* ------- *)

{ For Edit1 and Edit2 }
procedure TZipWnd3.Edit1DragOver(Sender, Source: TObject;
                                 X, Y: Integer;
                                 State: TDragState;
                                 var Accept: Boolean);

begin
  Accept := True;
end;


type  ExtStrT = String[49];
const  InclExtStrS : array[0..4] of ExtStrT =
       ('*.*', '*.txt',
        '*.dpr *.opt *.res *.dcu *.dfm *.pas *.txt',
        '*.bmp', '*.pas');
const  ExclExtStrS : array[0..3] of String[9] =
       ( '*.exe', '*.~dp', '*.~df', '*.~pa' );

{ For Edit1 and Edit2 }
procedure TZipWnd3.Edit1DragDrop(Sender, Source: TObject;
                                 X, Y: Integer);
var  txtS : String;
begin
  if  FiBox2.GetDirItem(txtS,Source)  then
    if  (Sender as TEdit).Name = 'Edit1'  then
```

(ต่อ)

ลิสติงที่ 5 (ต่อ)

```
      begin
      if  tsSelExt.TabIndex = 0  then
        txtS := txtS + '\' + InclExtStrS[rgInclExt.ItemIndex]
      else
        txtS := txtS + '\' + '*.* -x' +
                        ExclExtStrS[rgExclExt.ItemIndex];
      end
    else  txtS := txtS + '\';
  (Sender as TEdit).Text := txtS;
end;

procedure TZipWnd3.rgInclExtClick(Sender: TObject);
begin
  Edit1.Text := Edit1.Text + ' ' +
                 InclExtStrS[rgInclExt.ItemIndex];
end;

procedure TZipWnd3.rgExclExtClick(Sender: TObject);
begin
  Edit1.Text := Edit1.Text + ' -x' +
                ExclExtStrS[rgExclExt.ItemIndex];
end;

(* ---------- *)

type Str5 = String[5];
const  OptAddS : Str5 = 'afu';
       OptMthS : Str5 = 'nxfs0';

function  TZipWnd3.GetOptStr : String;
var  extS : String[9];  cnt : Integer;
begin
  extS := '';
  with  gbMisc  do
    for  cnt := 0 to ControlCount-1  do
      if  TCheckBox(ControlS[cnt]).Checked  then
        extS := extS + Char(TCheckBox(ControlS[cnt]).Tag);
  Result := ' -e' + OptMthS[rgMethod.ItemIndex+1] +
            extS +  OptAddS[rgAdd.ItemIndex+1] + ' ';
end;

procedure TZipWnd3.bttOKClick(Sender: TObject);
var    cmmLneS : String;
begin
  cmmLneS := zipBatFiNmeS + ' ' + Edit2.Text +
             GetOptStr + Edit1.Text;
Label3.Caption := cmmLneS;
  Form1.Memo1.LineS.Clear;
  if  RunBatFi(cmmLneS, ZipLogFiNmeK, oRprtStrS)  then
end;


procedure TZipWnd3.CheckBox1Click(Sender: TObject);
begin
  if  CheckBox1.Checked  then
    if  MessageBox(Handle, 'Do you really want to delete' +
                        #13#10 + ' source files', 'Verify',
                        mb_IconStop or mb_YesNo) = id_No  then
      CheckBox1.Checked := False;
end;

end.
```

1. ในการกำหนดกรอบ Include Ext (rgInclExt) และกรอบ Exclude Ext (rgExclExt) ให้ดำเนินการดังนี้

- กำหนดกรอบ NoteBook ได้เป็น nbSelExt ตรงตำแหน่งที่เห็นเป็นกรอบ Include Ext ในรูปที่ 3

- ให้ค่าพรอพเพอร์ตี้ส์แก่ nbSelExt โดยเฉพาะค่า Pages ที่ให้เป็น Include กับ Exclude จะเป็นชื่อหน้า

- เลือกหน้า Include (โดยให้ค่านี้แก่ ActivePage) แล้วกำหนดกรอบ Include Ext (rgInclExt) และให้ค่าพรอพเพอร์ตี้ส์

- เลือกหน้า Exclude แล้วกำหนดกรอบ Exclude Ext (rgExclExt) และให้ค่าพรอพเพอร์ตี้ส์

2. ป้อนโปรแกรมทั้งหมด ซึ่งได้กำหนดให้ Edit1 และ Edit2 ใช้กิจกรรม Edit1DragOver และ Edit1DragDrop ร่วมกัน (ป้อนโปรแกรมได้โดย เลือกกรอบ Edit1 กดคีย์ Shift พร้อมเลือกกรอบ Edit2 เลือกหน้า Events และดับเบิ้ลคลิ๊กในกรอบทางขวาของหัวข้ออีเวนต์ OnDragOver และ OnDragDrop)

3. ในยูนิต PKMain1 ของหน้าต่างหลัก ให้บ่งใช้ยูนิต UZipWnd3 ใน uses และป้อนโปรแกรมของกิจกรรม Zip1Click

คำอธิบาย

ได้จัดโปรแกรมไว้สำหรับเรื่องต่าง ๆ ดังนี้

1. ไฟล์ .bat เพื่อตัดปัญหาในเรื่องไดเรคทอรีของไฟล์ . bat (และเช่นเดียวกันกับในหน้าต่างอื่นตามที่จะกล่าวต่อไปด้วย) ได้ให้

- สร้างไฟล์ dozip.bat ทุกครั้งที่รันโปรแกรมนี้ ใน FormCreate ด้วยกิจกรรม rgReportClick และลบใน FormDestroy

- เช่นกัน เมื่อผู้ใช้เลือกปุ่ม Screen หรือ File จะมาทำงานที่กิจกรรม rgReportClick ซึ่งจะสร้างไฟล์ dozip.bat ใหม่ ให้แสดงรายงานของโปรแกรม PKZIP ทางจอภาพหรือเก็บไว้ในไฟล์ ziplog.txt แล้วแต่กรณี

2. รับการดร็อป จะต้องจัดกิจกรรมให้แก่คอนโทรลที่รับการดร็อป 2 กิจกรรม และสำหรับของหน้าต่างนี้คือ

2.1 Edit1DragOver เพื่อตรวจว่าได้แดร็กมาจากต้นทางที่ถูกต้องหรือไม่ ของหน้าต่างนี้จะแดร็กมาจากหน้าต่าง File Box เท่านั้น จึงให้ Accept เป็นจริงเสมอ

2.2 Edit1DragDrop เพื่อนำค่าที่แดร็กมาใช้ โดยเรียกใช้ GetDirItem กล่าวคือ

- ถ้าแดร็กมาจากกรอบรายการไดเรคทอรี (GetDirItem เป็นจริง) และถ้าดร็อปที่กรอบ Edit1 จะต่อสตริงด้วยไวล์การ์ดตามแต่ผู้ใช้จะเลือกหน้า Include หรือ Exclude ไว้ แต่ถ้าดร็อปที่กรอบ Edit2 จะต่อสตริงด้วยเครื่องหมาย \

- แต่ถ้าแดร็กมาจากกรอบรายการไฟล์จะนำชื่อไฟล์มาใช้โดยตรง

- โดยให้ค่าแก่กรอบ Edit1 หรือ Edit2 ตามแต่จะดร็อปที่กรอบใด

3. โปรแกรมของปุ่ม OK การเขียนโปรแกรมของหน้าต่างนี้เขียนอย่างไดอะล็อกบ็อกซ์คือ ทิ้งค่าที่ผู้ใช้เซต (ป้อนหรือเลือก) ไว้ที่คอนโทรล จะรวบรวมต่อเมื่อผู้ใช้เลือกปุ่ม OK จะมาทำงานที่กิจกรรม bltOKClick ซึ่งได้ให้รวบรวมค่าเป็นค่าคอมมานด์ไลน์ให้แก่ cmmLneS มีค่าต่าง ๆ ตามลำดับดังนี้

- zipBatFiNmeS ชื่อไฟล์ dozip.bat

- Edit2.Text ชื่อไฟล์ .ZIP

- ค่าออปชัน โดยให้ GetOptStr รวบรวมจากกรอบต่าง ๆ (ซึ่งส่วนหนึ่งทำเป็นค่าคงตัวสตริงไว้ สำหรับค่า Tag ของกรอบ CheckBox คือค่ารหัส ASCII)

- Edit1.Text ชื่อไดเรคทอรีหรือชื่อไฟล์ ที่จะนำมาย่อ

เสร็จแล้วได้ให้ RunBatFi รันไฟล์ dozip.bat ตามค่าในคอมมานด์ไลน์ และได้ผ่าน

- ชื่อไฟล์ ziplog.bat ใน ZipLogFiNmeK เพื่อเก็บรายงานการย่อไฟล์หากต้องใช้

- ออปเจ็กต์ของสกุล TStringS ใน oRprtStrS ซึ่งกิจกรรม rgReportClick จะเป็นผู้ให้ค่าเป็น nil เมื่อให้แสดงทางจอภาพ และเป็น Form1.Memo1.LineS เมื่อให้เก็บลงไฟล์ ซึ่งจะเป็นการนำออกแสดงตามที่กล่าวมาในหัวข้อการรันไฟล์ .bat

สำหรับโปรแกรมนอกนั้นเป็นไปตามที่กล่าวในหัวข้อวิธีใช้

ท้ายตอน

ฉบับนี้คงได้แค่การย่อไฟล์ การขยายไฟล์ต้องขอเป็นฉบับหน้าครับ สำหรับโปรแกรมผมได้ฝากมาทางซอฟต์เมล์ทั้งหมด เป็นโปรแกรมที่ผมเขียนขึ้นใช้เอง นำมาปรับปรุงบ้าง แต่ก็ยังมีข้อบกพร่อง

แก้ที่ผิด

ในบทความ "โปรแกรมด้วย Delphi : เมนู" ในไมโครฯ 126 ตกการให้ค่า Tag ขอให้เพิ่มการให้ค่า Tag แก่รายการ Cut, Copy, Paste, Delete และ Select All เป็น 2, 3, 4, 5 และ 6 ตามลำดับ โดย

1. ดับเบิลคลิ๊กที่ MainMenu1 (ที่ไอคอนของคอมโปเนนต์นี้) เพื่อเปิดเมนูเอดิเตอร์

2. เลือกรายการแล้วให้ค่า Tag

ขออภัยครับ